Metamorphosis

Into a New Era

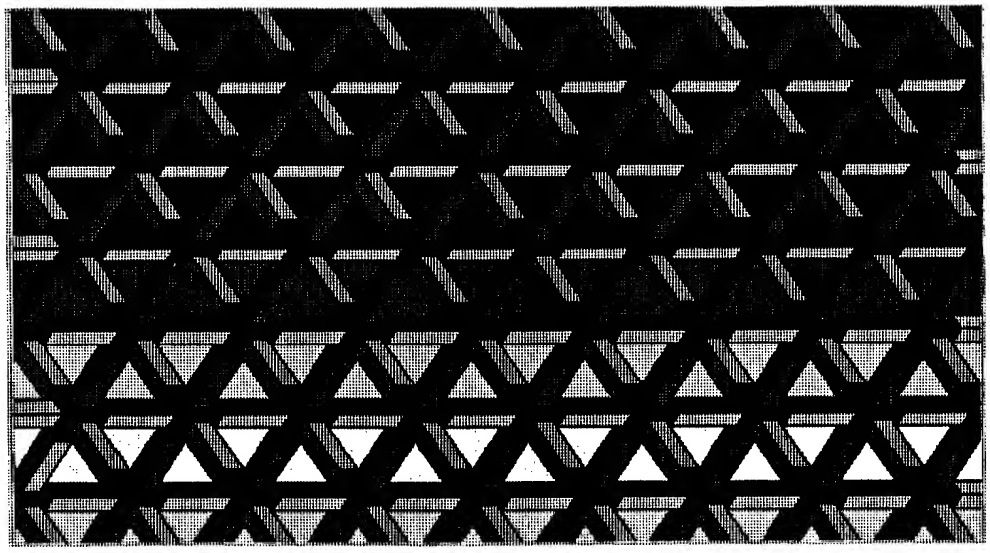
Vol 1, Nbr 6 | For the CoCo/OS9/OSK Communities | November 1993

Beginning a Complete OS-9 File System Tutorial

Part V of The Art of Programming

Return of the Hardware Hacker!

Review: Speedisk



November 1993, Vol 1, Nbr 6

Contents

Metamorphosis
Published monthly by
The Dirt Cheap Computer Stuff Company
1368 Old Highway 50 East
Union, Missouri 63084

Publisher
Mark D. Griffith

Editor
Barbara Ann Griffith

Contributing Writers
Mark Griffith
Mike Guzzi
Shawn Marolf
Tim Wilhite

Subscriptions
Metamorphosis is published 12 times a year. Subscription rates are $24 per year for the continental United States. Canadian rates are $32 (US) per year. Overseas rates vary. Please write for details.

How To Contact Us
Mail can be sent to the address shown above. Electronic mail can be sent via the Internet to:

76070.41@compuserve.com
or
MARKGRIFFITH@delphi.com

Telephone calls can be made to (314) 583-1168. Please leave a message and your call will be returned.

This publication is composed, formatted, and master pages created entirely on machines running the OS-9 operating system.

Features

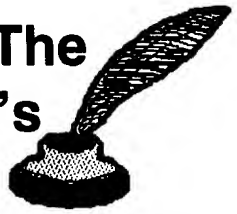
Columns

Departments

On the Front Cover

This image represents what a badly structured disk may look like. With the file structure tutorial beginning in this issue and the new disk defragmentation program available, your disk should never look like this! The image is a 640 x 350, 4 bit per pixel GIF file. It was converted to a Postscript file on an MM/1.

From The Editor's Desk



Ah, the month of November. It's a wonderful month for making plans for family gatherings. The air is generally invigorating and crisp, filled with scents of the surrounding countryside. (In our case this means burning firewood, damp leaves, and mud.) It's definitely time to drag out those warm, comfortable sweatshirts, sweathers, and blankets . . . lots and lots of blankets. November, and the holiday season it initiates, is a time for reminiscing about all those wonderful family members that we somehow just haven't managed to get over to see during the past year. November is just full of comfortable, cozy clothes and memories.

November is also a wonderful month for eating. All month long you plan for and buy for and look forward to eating golden brown turkey, pumpkin pie with whipped cream, cranberry sauce, giblet gravy, sweet potatoes, savory stuffing, and all this to be followed with indigestion remedies. (You know you "really" enjoyed your Thanksgiving dinner when you waddle away from the table rubbing your belly and asking for the Alka-Seltzer, DiGel, Tums, and any and all other indigestion medicines in the house!) We hope all of our subscribers (and yes, even those that haven't joined us yet) had a very wonderful Thanksgiving holiday, and that your tummies didn't have to pay too high a price afterwards. Hopefully, everyone had a nice, comfy place to curl up on the sofa after dinner.

We decided to splurge a little on Thanksgiving dinner and had a table full of food. Times have been lean lately which made it extra special to be able to sit down to a bountiful table, even if it was only going to be for one or two days. Thanksgiving was quite satisfying for us this year. We even got a little bit of the white stuff (snow!).

We recently had the pleasure of visiting with Tim Jones. He was in the St. Louis area on business and decided to come by to see our set up for the magazine and Dirt Cheap Computer Stuff. We had a real nice visit with Tim and would like to extend the same offer to our other subscribers. (He even survived my cooking!) If you're planning to be in the St. Louis area give us a call. We'd love to have you over for dinner and a visit. (I have a new stove and am not burning dinner now!) Tim even survived meeting Buck (our shepherd-mix male dog), Penny (our dachshund), and Samantha (the stupidest excuse for a dog I have ever seen).

It seems to me that there are a lot of people in the OS9 community that few people have actually met. Mark will tell me about someone, I ask what he looks like, and Mark says he's never met the guy, but he's conversed with him over the network for such and such number of years. Since this happens all the time, I thought it might be a good idea to feature a person or family in each issue. I, for one, would like to get to know a little bit about the people in the OS9 community. This could be sort of a "Let's get acquainted" kind of column. If anyone is interested, please send me a brief autobiography. Helpful information might include name, work and educational history, technical experience/knowledge, hobbies, etc. Photos are hard to reproduce, but if you include one we will see what we can to do get it to look good. (Let us know if you would like for us to return it.)

Christmas is just around the corner. A little boy on my bus told me the other day that Santa had a lean year this year and probably wouldn't be bringing him many toys. He told me that was all right though, because he really didn't need that many new toys. It's amazing how you can tell which thoughts are from a child and which are from the parents. Times are lean all over. I sure hope 1994 is better than 1993.

A quick apology for the printing quality in the last issue. Many of you received copies that were very poorly done. We didn't notice it until we started putting them together, and then it was too late. Our printer has been notified and assures us this will not happen again.

Concerning our lateness in getting each issue out. We must apologize to our readers. We had hoped to be caught up before the end of the year, but it doesn't look like that will happen. There are a number of reasons, mostly due to Mark being so busy both at home and at work. However, our problems do not justify our lateness, just explains it.

We want everyone to know that your full subscription will be honored. No one will receive less than twelve issues. If that means going beyond October or November next year, then so be it. We will continue to work hard to catch up, hopefully early next year. We will also strive to improve the overall quality of the magazine.

Keep those letters coming. We enjoy hearing from everyone. We hope you enjoy hearing from us, too!

Mail Call!

Subscriptions Keep On Coming!

Hi Mark

I'm about to send (finally!) a check for a year's sub to your mag. Was wondering if it'd be possible to start my sub with the first issue following the freebies? Last issue I have is Vol 1, Nbr 4, September 1993 so I guess that would mean starting with the November issue. This isn't a major consideration, it's just that finances have been tighter than I like, lately, so I'm late to send you money, but I hate to miss an issue.

Roy Smith

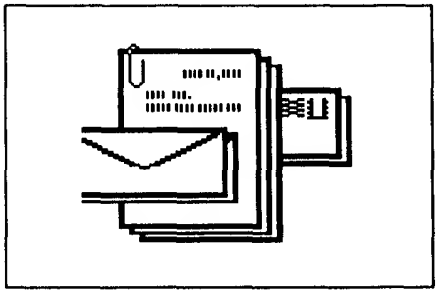
BBS List Corrections

Mark and Barbara,

I certainly enjoyed reading your latest edition of Metamorphosis (formerly No Name Magazine). I am looking forward to the next year and my new subscription. Thanks for your help at the Atlanta CoCofest. Your BBS LIST has several errors that need to be corrected. First, the ACS BBS has been listed twice. The listing with ACS capitalized is the correct/preferred one! You also list ACCUG but that BBS has been defunct for quite some time and that telephone number no longer belongs to Phil Zeigler. I am not surprised to see the state of Washington so well represented; there are many active CoCo users in that area. Lastly, your listings that contain BBS's operated by our Canadian cousins is not uniform in the way it presents the location of the BBS. Identification such as Surry, BC and Calgary, AB are preferred by me while Ontario, Canada is a bit vague to me.

Alan Dages
Atlanta, Georgia

We certainly do appreciate your taking the time out to help us keep our list straight, Alan. Your corrections have been noted and inserted in the list. Let's hope that next year's list will be larger and better than this one. It's only through the efforts of the readers that we can keep this list current. Everyone, please, help us out.—Barbara



Looking for a Screen Dump

Dear Barbara,

Someone mentioned over the CoCo Listserv the need for a screen dump program which works with the CoCo3 Hi-res screens in RSDOS for the CGP-220. I am also curious as to what example code would look like, since I have the need to create a filter in OS-9 to create color print outs on the CGP-220 from bitmap format. If you know of anyone who has such code, I would appreciate a quick example of it in any language, including RSDOS BASIC.

The CGP-220 has an interesting mode which makes the dumping of full color graphics complex, and I have yet to see working source code to make use of it.

David Halko

I certainly hope that one of our readers will have the information you need, David, and write to us. Thanks for your time!—Barbara

We Promote!

Mark,

Got the magazine today (actually yesterday) and read it. Very nice job. I have one complaint though. You promoted me. I am the Senior Vice President of the Glenside Club, not the President. Tony Podraza is and has been the President for some time now. I am the President of the Chicago Area OS9 Users Group though. I can understand the confusion. I was the liason for the Chicago CoCoFest. Anyway, it would be nice if you would print a correction in the next issue.

You also might mention that Brian Schubring has been elected President for 1994 (the Glenside Club). Thanks much. Keep up the good work!!! Now if you could only add about 500 pages so I'd have enough to read till the next issue. <<<GRIN>>> I have to content myself by reading rags like LAN TIMES and Electronic Media. Best wishes and a Happy Turkey Day.

Carl Boll

Sorry about that Carl! I'm glad you set us straight. Also,

thanks for the very kind words. We hope to give you what you want. Look for some really good articles coming in the near future, and more pages too!—Barbara

News Clips

Rumor Mill Central

Certain well known OS-9 persons have had some thoughts about building a new OSK machine geared toward the CoCo3 users. Like the original concept behind the MM/1, it will be designed to move CoCo3 OS-9 Level II users into the OSK world. The initial idea is to make a cheap mother board with several IBM-AT type expansion slots, SIMM memory sockets that can use 256K through 4 Meg or higher SIMMs, and at least one serial port and possibly one parallel port. This new machine would then have drivers written to support VGA and SVGA graphics cards, multiport serial cards, ethernet cards, and others. The best part of all is the initial price for a mother board with OSK would be around $399.00 with no memory installed. VARs could then offer package deals to the users that would include cases, keyboards, monitors, and so on.

This is also along the lines of the original CoCo....a cheap initial outlay with the ability to expand it in bits and pieces as the user saved enough money. Would a computer like this be in demand? The builders would like to know before they begin. Please let them know by writing to us here at *Metamorphosis*. We'll make sure they see your comments. Don't forget you can write to us via the Internet, Compuserve, or Delphi.

UUCP for OSK!

For some time, actually a year or more, Mike Haaland has reportedly been working on a completely new port of UUCP for OSK. However, because of all sorts of time consuming events that have happened in Mike's life, he has been unable to finish the port.

Meanwhile, Bob Billson announced that he had recently finished some hacks to Rick Adam's UUCP for OS-9/6809, and called it UUCPbb. Then, Boisy Pitre announced that he has already ported UUCPbb over to OSK, then someone said they had Elm ported, then someone else said they had Palm ported, and the list goes on.

So it seems that the OSK world will soon have a real UUCP port to work with thanks to the CoCo3 people and a few industrious souls. A well round of applause goes to all these people for their dedication!

Return of the One-Liners

Because of the work of Rodney Hamilton, a favorite item from the old *Rainbow* days is again available to our CoCo readers. Rodney has donated his personal collection of one-liners to be printed over the next several issues. I always looked first for the one-liners when each issue of the *Rainbow* arrived. Thanks to Rodney, this trend will continue for some time.

If there are any readers that have more one-liners they would like to send in, please do. I'm sure everyone will enjoy them!

Support your local CoCo Club

The **Color Computer Owners Group** is an active club in the Detroit metro area. They are actively seeking new members who are interested in CoCo only activities. The CCOG also supports a local BBS, **Jim's CoCo Corner.** This BBS has been online since 1987 and provides a message board full of CoCo help of all kinds. This BBS is also a contact point for the CCOG. Give it a call at 313-292-4713 24 hours a day 7 days a week. Jim's Corner supports 300, 1200, and 2400 baud, 8/N/1. You can also contact the club secretary, Robert Gault, at 832 N. Renauld, Grosse Pointe Woods, MI 48236

The MM/1 Lives Again

David Graham recently announced that his company, **Blackhawk Enterprises,** has bought the rights from IMS to continue making MM/1 computers. The details were not released, but inside sources say that Blackhawk received all the rights and remaining OSK licenses to produce at least 50 more MM/1's immediately. A large number of completed MM/1 board sets are still at the manufacturer and all David needs are some orders to break them loose. With the recently released MM/1 accelerator board and the new software that is becoming available, it looks like the MM/1 might still survive for a couple more years. Good luck David!

Mark Griffith

The Art of Programming

Part V

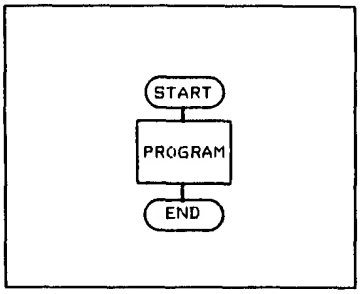
Programming Structure

by Shaun Marolf

Shaun Marolf, 30, is a member of the US Navy stationed at Naval Air Station San Diego where he is currently working on his degree in Computer Sciences. He first learned programming in 1979 on an Apple II Plus and is also schooled in electronics and digital circuits. Shaun runs the "Eight Bit Heaven" BBS (619-447-2111) and owns several computers, including an original grey case CoCo 1.

The physical structure of a program will largely depend on the language you use. We will cover the two types of structures most used today. Both intertwine with each other.

The first structure is the top to bottom or straight through structure.



Top to Bottom Structure

In the top to bottom structure, the entire program is put together into one block, and the entire program is loaded into memory. The advantage of this structure is that it has a predefined logic order in which it operates. This allows for the programmer to have a much greater control over execution and data flow.

Disadvantages are that any changes made to the program can and usually do affect the rest of it. The larger the program, the harder it is to make changes. Also, the program may require a large chunk of memory to hold it. Care must be taken to ensure that enough memory is left for the data.

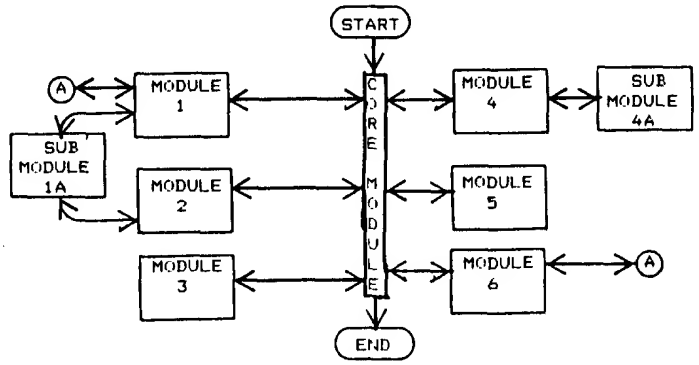
The modular programming structure has a core module which controls I/O and data flow. It accesses other modules that manipulate data, which may in turn access sub-modules not linked to the core, or even other modules. Modules may also access each other.

The largest advantage to this is that only the core module need be maintained in memory. Other modules are loaded and unlinked as needed. This method requires less memory for execution and allows more for data storage and manipulation. It is also easier to make changes since the modules are separate programs themselves.

The disadvantages of modular programming are that the logical program and data flow are now only the programmer's responsibility. Care must be taken that the wrong module is not loaded. If the core calls the wrong module or a module uses a sub-module or calls on another module, that disturbs the correct logic flow. The program can then bomb out or the data can be altered in such a way that it becomes useless.

Though changes are easier to make, any changes made within a module will likely affect other modules. The core is the most likely one to be affected. Any changes made in a module must also be reflected in the other modules it affects.

The type of structure used should be carefully considered and decisions should be made



The Modular Structure

based on the program's overall functions. If the program is in BASIC, it is most likely that you will use the top to bottom structure. You can use the modular structure too, but it requires knowing the tricks of BASIC code. C, on the other hand, is designed for the modular programming method but can be easily done in a top to bottom format. Since the top to bottom structure is extensively covered by others, I won't discuss how to organize this format. Next month I'll cover how to organize a modular programming structure.

From The Jargon File

Code Grinder *n.* - 1. A suit-wearing minion of the sort hired in legion strength by banks and insurance companies to implement payroll packages in RPG and other such unspeakable horrors. In his native habitat, the code grinder often removes the suit jacket to reveal an underplumage consisting of button-down shirt (starch optional) and a tie. In times of dire stress, the sleeves (if long) may be rolled up and the tie loosened about half an inch. It seldom helps. The *code grinder's* milieu is about as far from hackerdom as you can get and still touch a computer; the term connotes pity. See *Real World, suit.* 2. Used of or to a hacker, a really serious slur on the person's creative ability; connotes a design style characterized by primitive technique, rule-boundedness, *brute force,* and utter lack of imagination. Compare *card walloper;* contrast *hacker, real programmer.*

OS-9 File Structure

Part I

by Mike Guzzi

Mike Guzzi, 25, has a BS in Electrical Engineering, and is currently a technician at Specialty Records Corp/WEA Manufacturing INC. Mike wrote the APBBS BBS system that was sold by Second City Software. He has made several contributions to the public domain including Mbackup which allows a CoCo user to use more then 64k for copying a disk legitimately, under OS-9. Mike has owned a CoCo since 1984. He can be reached at **mike_guzzi@delphi.com.**

Over the years working with OS-9, I have taken for granted that long ago I have conquered this concept. Moving around the OS-9 file structure seems second nature now, but I remember a time where this wasn't so. Fortunately we have had some books written on the subject and a few tutorials but still some have a hard time understanding it. I remember back when I was in college I was involved in a color computer club. One night we were discussing OS-9, and the majority of people just didn't understand how to navigate OS-9. Fortunately, we had a CoCo3 nearby so I booted OS-9 and started moving around the disk. Unfortunately, many people just couldn't see what I was doing. I then asked for some paper and a pencil and started drawing out what an OS-9 disk looked like. I used that in conjunction with the computer to show exactly what I was doing and where I was. The club members were actually able to see on paper what I was doing on the computer. So, with recent inquiry on this I decided to recreate this in this file and hope that it will help you understand how to navigate your OS-9 system.

Most people start off using OS-9 on a floppy based system. The bare minimum hardware to run OS-9 Level I is a 64K Color Computer with 1 disk drive. If you're running OS-9 Level II it requires a 128K Color Computer 3 with 1 disk drive. Unfortunately with minimal hardware it does tend to complicate things, unlike Disk Extended Color Basic, OS-9 stores a lot of its information on disk and will tend to cause confusion. I will start off saying that OS-9 is much better to operate if you own two disk drives. A hard drive will greatly increase OS-9's power since everything can be made available to OS-9. I have run OS-9 on a hard disk for years now, but many people now own hard disks and need to understand more about how to make OS-9 work properly and to minimize errors. First I will explain what makes OS-9 unique.

OS-9 is unlike many other operating systems. What OS-9 does is provide an environment for programs to run. It takes care of reading the keyboard, putting characters on a screen, drawing graphics, talking to your printer and modem. It also handles your disk drives. OS-9 is known as a modular operating system, this means it's not one big program that "does it all". Rather, it's a collection of small programs that work together to make up OS-9. Let's take a look at this. Boot up OS-9 according to the instructions in the manual. When you see the OS-9 prompt type the following in.

```
+--+--> You will see this
|  |
|  | +--+--> what you type
|  | |  |
OS-9: mdir
```

It will generate a listing similar to this

```
Module Directory at 16:50:43
REL       Boot      OS9p1
OS9p2     IOMan     Init
Clock     RBF       dd
CC3Disk   D0        D1
D2        SCF       ACIAPAK
T2        PRINTER   P
CC3IO     GrfInt    Term
W         W1        W2
W3        W4        W5
W6        W7
PipeMan   Piper     Pipe
GrfDrv    Shell     Load
MDir      Copy      Date
DeIniz    Display   Echo
Iniz      Link      Mfree
Procs     Rename    Tmode
Unlink    Xmode     Dir
```

NOTE: This listing will not match to yours. I took this off my system.

These are a collection of small programs that work together to make up OS-9. The first part of OS-9 is the kernel. When you type DOS from Disk Basic and you see OS9 BOOT on your screen, that is the kernel of OS-9. It is responsible for loading in the rest of OS-9, and managing it's basic functions. The first line makes up the OS-9 kernel (REL, Boot, OS9p1). Those modules are loaded when you type DOS and you see the OS-9 BOOT message on your screen. The next set of modules make up the basics of OS-9. I am presenting this in case your wondering what all these modules do.

OS9p2	This module handles things like memory, system calls, and such.
IOMan	This module handles all input and output, Input and Output can be described as things like typing on a keyboard, that is input to the computer. When you see things on your screen, thats output.

init	This module contains only data with constants for the system. This includes things like name of main window, default directories and default disk drive.
Clock	This module is responsible for providing multi-tasking. It also keeps the time of day.
RBF	This is called a "file manager module" and tells OS-9 how to deal with devices that store data such as floppy drives, hard drives and ram disks. All of the OS-9 file structure information is kept here. This way no matter what kind of disk drives you own, the file structure will always be the same.
SCF	This is also a "file manager module." This manager tells OS-9 how to deal with serial devices. These are devices that input or output 1 character at a time. Examples would be a printer, modem, keyboard and your display screen.
Pipeman	This is also a "file manager module." This manager tells OS-9 how to set up "pipes" between programs. This allows the output of a program to become the input of another. When you type something like this: `list bootlist | OS-9gen /d0` What this does is list a file called "bootlist", but instead of displaying it on the screen, it will send the information to a program called "OS-9gen" the '!' is called a pipe.
CC3Disk	This is called a "device driver" and tells OS-9 how to operate your floppy disk controller. It contains information on what information must be passed to the disk controller to tell it what to do.
D0	This is called a "device descriptor." This small module contains information describing the first disk drive attached to your controller. This holds information such as number of tracks, how many sides, and how fast to step the heads. It needs CC3Disk to operate.
D1	This is also a device descriptor. It contains information describing your second disk drive attached to the controller.
CC3IO	This module tells OS-9 how to get information from the keyboard, how to read your joysticks and mouse. It also tells OS-9 how to produce sounds through your monitor's speaker.
ACIAPAK	This module, also a device driver, tells OS-9 how to interface with an RS-232 pack plugged into the Multi-pak expansion.
T2	This module, a device descriptor, describes the serial port of the RS-232 pak. It contains information such as baud rate, parity, word length and so on.
GrfInt	This is a system module. This module contains information on how to do graphics. It holds information such as drawing lines, boxes how to change colors, and other windowing commands.
Term	This is a device descriptor which describes to OS-9 the first window (display screen) you see when you boot OS-9. It contains information like what kind of screen it is (text or graphics) the physical size of the screen, colors, and so on.
W1-W7	These are other windows (display screens) you can set up to have additional screens to run programs. These modules is what allows you to have different programs running in different display screens at the same time.
PRINTER	A device driver that tells OS-9 how to interface to the serial port on the back of the computer to use it to drive a printer.
P	A device descriptor that holds information about the printer. It contains information like baud rate, linefeeds, word length, and so on.

GrfDrv — This is a module that works with GrfInt. This module contains information on how to manipulate the hardware inside the computer to provide you with the screen display.

Shell — This is a program that is the interface to OS-9. When you see the OS-9: prompt, that is this module. It takes what you type as a command and tells OS-9 to run it.

You will also notice names like "dir", "mfree" and so on. These are utilities that are loaded into memory. You can think of mdir as a directory of what's stored in memory.

It may seem confusing to have several modules just to talk to your floppy disks but consider this. You can have up to three disk drives attached to the same disk controller. You only need 1 module (CC3Disk) to talk to the controller, and then three modules to describe each disk drive (D0, D1, D2). This saves memory since the controller is the same for each disk drive, and yet each disk drive can be different, which is why you need a small module to describe each disk drive. An example of this could be a setup where your first disk drive is a double sided 40 track disk drive, and your second one can be a double sided 3.5 inch disk drive. This allows greater flexibility in how your system can be set up. For example, on my system I have four serial ports. I have the module ACIAPAK which tells OS-9 how to talk to the RS-232 pak. There are also four modules, T2, T3, T4, T5 which describe each port on the pak. I can set one port to 2400 baud for my modem, a second to 9600 baud for a terminal, and so forth.

Ok, you may be wondering why I went through that. Well, when you want to get to your files on disk, or send information to your printer, you must tell OS-9 what device to use. Under OS-9 all of your input and output devices are considered files. There are two types of devices (and therefore files) that OS-9 will work with. The first is called "random block files" which the module RBF handles. These are devices where you can store and retrieve information on things like disk drives, hard disks, and so forth. You can read or write information anywhere on these devices. They hold multiple files, and you can access them anyway you want. For example, let's look at your system disk.

```
directory of /d0   11:18:27
OS-9Boot     CMDS         SYS      startup
window.t38s  window.t80s  window.glr4
```

You can choose what file you want to use. Look at the files with the name window in it. Each file is stored on a different part of the disk, but you can access any of them in any order that you want.

The second type of file is called sequential. These are files that input or output 1 character at a time in one long data stream. If you were to type all the letters of the alphabet from your keyboard like this:

```
OS-9:abcdefghijklmnopqrstuvwxyz
```

Each keystroke was read one at a time by OS-9. When you send data out to your modem, the modem takes the information one character at a time. You can also read information from your modem, but only one character at a time. This will become more clear and important next month when we discuss pathlists and files. See you then!

Crunching Your Hard Drive

Speedisk Disk Optimizer for OSK

by Tim Wilhite

SPEEDISK is a fully featured disk optimizer for OSK written by Brian White of *Dynamic Systems*. It first became available just prior to the October Fest in Atlanta this year. I had the priviledge of being a beta tester and have seen SPEEDISK grow and improve over several months prior to its release. The version used in this review is another beta version with added support for termcap capabilities. More on this later.

SPEEDISK comes on one 3.5 inch floppy drive along with a very small user manual. This is not to say that the docs are incomplete. It just means that there is not much to running SPEEDISK—it takes care of everything itself. I understand that another version that contains a greatly enhanced document is being worked on for commercial users. I have not seen this so I can't comment further.

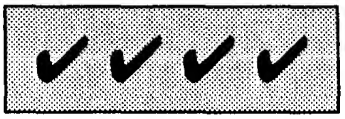
Installing SPEEDISK is simplicity itself. All you need to do is copy the SPEEDISK program from the floppy disk into your commands directory and then type **speedisk.** By default, SPEEDISK will begin an analysis on your /dd device unless you give it a device name on the command line like **speedisk /h2.** SPEEDISK will also work with other media such as floppy disks, Floptical disks, and Syquest cartridges.

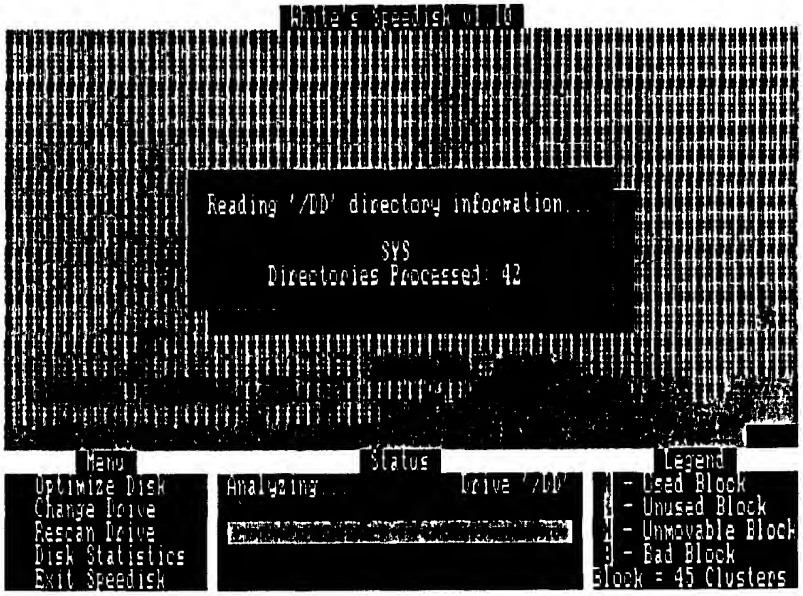
I must point out that it is critical that there be no other disk access while SPEEDISK is running. Personally, that was not a problem. I did not have any files trashed, and I have not heard of any problems with SPEEDISK trashing files for other testers. However, the prudent user will want to make sure just to be on the safe side.

I did not have any trouble when using SPEEDISK in this fashion. There is no mouse interface, but the keystrokes are straight forward in running the program. The first letter of the function will start the ball rolling.

Once SPEEDISK begins, it will read your entire hard disk structure looking for errors. SPEEDISK **will not** continue with optimization if any problems have been found with the disk. This is a very nice safety feature put in by Brian. Once it has completed checking the disk, SPEEDISK will then display the following statistical information.

Directory Fragmentation
File Fragmentation
Free Block Fragmentation
Disk Utilization
Total Space on Disk
Nbr of Directories
Nbr of Files
Nbr of Hard Links





Bytes in Directory
Bytes in Files
Bytes for System Use
Bytes in Bad Blocks
Bytes Inaccessible
Bytes Free

Once SPEEDISK has checked the drive, you can have it continue on to optimize the hard disk or not, rescan the drive, change the drive, or exit the program. I believe all of these functions are self-explanatory.

At this point, a small flaw shows up. The hard drive device descriptors distributed with my MM/1 have the control byte set so the drive cannot accidentally be formatted. By using the **dmode** utility to change the control byte from $000b to $000a, you can disable this feature. This is required to allow SPEEDISK to optimize your drive. On my MM/1, device descriptors named "h0fmt" or "h1fmt" are supplied with the control byte set to allow disk formatting. To use that device, you'd have to load it in advance and then give that device name on the command line, such as **speedisk /h0fmt.** I'm sure this same "problem" would exist on other OSK machines. This is a small problem, but I would like to see Brian incorporate an option in the future to automatically set the correct bit in the device descriptor. Invariably, I would forget to change it before I used SPEEDISK.

The graphics displayed while SPEEDISK is running are similar to the Norton Speed Disk utility and are shown in the screen capture on the previous page. Pound signs (#) are used to denote used blocks and, on the MM/1 display, a blank denotes an unused block. On the termcap display, a period denotes an unused block. While SPEEDISK optimizes the drive, you can see the blocks move around as they are read and written. Also used is a status bar which gives you the percentage of the drive completed. The display is updated very quickly and, while it shows you everything the program is doing, it is too fast to make much sense. But it looks nice and at least you know it is doing something other than just sitting there.

SPEEDISK also does some nice things like move all the directory files to the beginning of the disk to optimize searches. It also moves any bootfiles it finds to the end of the disk. SPEEDISK also has a verify feature that will verify all disk writes. This, of course, slows down optimization considerably. Command line options are also included to allow SPEEDISK to be run in "batch" mode from CRON or a user script. This allows you to optimize your drive at night when system usage is at a minimum.

SPEEDISK was written to be as bullet-proof as possible. During the optimization process, it tries to keep on files on the disk as it moves them around. If something happens, like a power outage, you can still recover. SPEEDISK includes a command line option to recover from just such a failure. Like the UNIX "fsck" program, SPEEDISK can be run over and over on the drive to clean it up. Once SPEEDISK gives it a clean bill of health, you can then proceed to optimize it again.

My hard disk is a 52 megabyte Quantum. Because the disk is so small, it only took SPEEDISK about 45 minutes to optimize the drive the first time I used it. With periodical use, I have cut the time to about 30 minutes. The author says that SPEEDISK optimizes the drive at about 1 to 2 megabytes a minute, which seems accurate. Of course, this figure depends a lot on how badly the drive is fragmentated and how full it is. Other users have reported similiar times for their drives.

The version of SPEEDISK I tested has two built in display systems. One uses the MM/1 graphics screen to draw the display and the other uses **termcap** functions to draw essentially the same display. I was not able to determine any real difference between the two except that the termcap display was slightly slower. SPEEDISK is smart enough to know when it is running on an MM/1, and will use the MM/1 screen display when it is. Otherwise, the termcap display is used. *(The screen capture shown is the MM/1 display—Ed.)*

I would give this program a nine and a half on a scale of one to ten. *(Which warrents four checks, our highest rating—Ed.)* It would rate a ten if I didn't have to remember to change the hard drive descriptor each time I used it. SPEEDISK is easy and safe to use. With these items in mind, I would highly recommend the purchase of SPEEDISK. I cannot give Brian White enough accolades for a job well done. The program is intuitive; it keeps you updated with what it is doing. Brian has given a great deal of thought to this program. SPEEDISK is certified to run correctly on just about ANY OSK system, from version 2.1 through 3.0. It has been in development for over three years so all the bugs have been worked out. This is a solid program and $99.95 well spent!

SPEEDISK is available from the Dirt Cheap Computer Stuff Co., 1368 Old Hwy 50 East, Union, MO 63084. 314-583-1168.

This one-liner provides a screen menu selection of up to nine one-liner programs. Each one-liners that uses "RUN" to re-start should be modified to use "RUN#" instead, where "#" is its specific line number to avoid going through the menu each time.

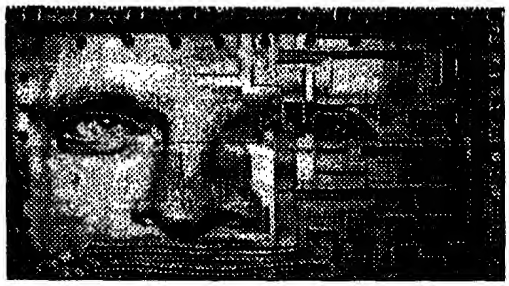
```
0 CLS:PRINT"ONE - LINERS"
:PRINT1"SCANNER",,2"CHARSET"
,,3"TINT ADJ",,4"MATH TEST",
,5"FILE DUMP",,6"CIRCLES",,7
"ANIMATION",,8"ROM->RAM"
:PRINT: INPUT"RUN LINE"; Z:
ONZ GOTO1,2,3,4,5,6,7,8:RUN0
```

The one-liners that appear in the menu will appear over the next few months.

Hardware, From the Inside Out

Testing Memory, Part 1

by Mark Griffith



Hardware hacking is something I have always been interested in. I've been an avid electronics buff since my early days back in the mid-sixties when I was tearing apart old TV sets and building ham radio gear. I remember transistors were just becoming commonplace and most of my stuff was still vacuum tube driven. My ham radio rig was mostly 2-meter experimental stuff....bouncing signals off meteor scatter and tracking the very first OSCAR satellite. I made the transmitter, a 15 watt AM unit, and the 2-meter frequency converter for my ancient (even for those days) Hallicrafter shortwave radio. Then came integrated circuits, computers, and....well, you know.

The purpose of this column, as indicated by the name, is to help show how a user can interface their OS-9 machines, be it a CoCo or a GIMIX 68040, to the outside world. I'll cover both the hardware and software aspects of this and through the coming months we'll develop some useful projects. I have all sorts of ideas for things we can cover. However, I don't always have the time to develop things when I would like too, so there will be months that I'll feature a guest columnist. So, if anyone out there would like to submit a pet project they are working on for a future column, please contact me at the magazine address or on the Internet, Compuserve, or Delphi.

This month we'll begin discussions on a task that many people can use. We'll also learn some techniques for programming that will help us in our future experiments. These techniques will then carry over into future columns.

Memory checkers have been around for many years. Almost everyone that has used a PC in the past 10 years has seen one in some form or another. Almost every boot sequence for almost all computers, large and small, include some sort of memory check to see if everything is OK before installing the kernel code. Memory checkers can be very simple, or extremely complex. Many articles in the technical journals have appeared over the years outlining different algorithms for checking memory. We'll start with a simple one here and move to more complex themes in the coming months.

Before you can check memory you have to have some way to access it. On an OS-9 machine, you can't check memory that is in use except for reading it to see if it can be accessed. This is usually a waste of time. If you had bad memory in use on an active system, you'd find out about it soon enough. The trick is to check the memory before installing it. There are a number of commercial stand-alone memory checkers on the market today, but they normally run several hundred dollars for the least expensive.

We're lucky. The MM/1 is rather unique as far as computers go. If you boot it as a 1 meg system, the expansion RAM is not used, but it *is* available and can be written to and read. With some simple code, you can use your MM/1 as a SIMM memory checker. This becomes more important to those people running the new accelerator boards or performing the 4 meg hack. You'll want to check out those new (or used) SIMMs before installing them on your system to make sure they work OK. If you installed some sort of zero-insertion-force (ZIF) socket to hold the SIMM memory, you could run through a bunch of SIMMs in little time.

Probably the simplest memory check is to see if the RAM is being refreshed. This also can check the refresh circuits of the computer, but it is more common for a certain bit or even an entire chip to have its refresh circuits go bad. Most dynamic RAM chips use tiny capacitors as the means for holding a bit in one state. Since capacitors loose their charge rather quickly (depending upon their rating in farads), these little capacitors need to be "refreshed" or zapped often so they don't loose their charge.

The refresh period is specified by the chip manufacturers and is normally somewhere around 12 milliseconds. This gives the maximum time the chip can reliably hold a charge without being refreshed. As long as the refresh pulse comes on a regular basis within that time limit, the chip will work correctly. However, if one or more of those little capacitors go bad, it will loose its logic state before the refresh period.

One way to test a chip to see if it can hold a "charge" is to write a pattern to memory, wait a little bit, and then read it back to see if it has changed. If it has, it would indicate that chip is bad. Of course, if you

get a bad reading from *every* memory address, then you might suspect the refresh circuits or some other component to be faulty.

The next type of memory test is the "walking bit" test. This is a very commonly used test that needs a little explaining. A single memory location in most modern computers is one byte, or 8 bits, "wide". In early machines, like the first IBM PCs, memory chips were normally inserted in groups of eight. For example, eight 16K chips gave you 16K of eight bit wide memory. Inserting eight 64K chips bumped you up to 64K, then 256K chips and so on. Today with SIMM memory, there are still eight chips used on one SIMM. So a one megabyte SIMM will normally have eight, one megabyte chips on it. Some use only two chips that have four bits each. Some SIMMs have nine chips on them, the extra chip used for memory parity checking.

To determine which of the eight chips on a SIMM or in a memory bank has gone bad, you need to check each bit of each memory location. This is what the walking bit test does. All you do is to write to a $01 to a memory location and then read it back. If that was successful, then shift the set bit over to the next chip and test it. Once you have tested each bit, then go on to the next memory location. As you might guess, this test can take a long time to test large amounts of memory.

The last type of memory test we'll explore is the shadow test. This is even more time consuming than a walking bit test since you basically do a test of each bit in each memory location, and then look through *all* the other memory locations to see if any other bit has been also set. This test will determine if memory address lines are shorted.

Now that I've explained what we'll be doing, next month we'll have some sample code to work with. From then on, we'll start exploring other areas of connecting your OS-9 computer to the outside world. I'm looking forward to writing this column and developing the projects. I'm sure this will be as much fun for you as it is for me.

Dirt Cheap Computer Stuff Co.
1368 Old Highway 50 East
Union, Missouri 63084

Address Correction Requested